Computer Engineering Department
Bilkent University

CS533: **Information Retrieval Systems**
Assignment No. 1 (No more progression: Done)
February 19, 2016
Last Update: February 25, 2016 - 6:30 pm
Due date: March 11, 2016; Tuesday, by class time (hardcopy is required)

Notes: Handwritten answers are not acceptable. The next assignment may overlap with this one. You have to solve preferably all at least any five questions in the order of question numbers.

1. Consider the following search results for two queries Q1 and Q2 (the documents are ranked in the given order, the relevant documents are shown in bold).

Q1: **D1**, D2, D3, D4, D5, **D6**, D7, D8, D9, D10.

Q2: **D1**, D2, **D3**, D4, D5, D6, D7, D8, **D9**, and D10.

For Q1 and Q2 the total number of relevant documents are, respectively, 10 and 6 (e.g., for Q2 three out of six relevant documents are retrieved).

a. Using the TREC interpolation rule, in a table give the precision value for the 11 standard recall levels 0.0, 0.1, 0.2, … 1.0. Please also draw the corresponding recall-precision graph as shown in the first figure of TREC-6 Appendix A (its link is available on the course web site).

Hint. "Interpolated" means that, for example, precision at recall 0.10 (i.e., after 10% of relevant docs for a query are retrieved) is taken to be MAXIMUM of precision at all recall points >= 0.10. Values are averaged over all queries (for each of the 11 recall levels). These values are used for Recall-Precision graphs

Please do this for each query separately and obtain one table for both queries using the average of two values at each recall point.

b. Find R-Precision (TREC-6 Appendix A for definition) for Query1 and Query2.

c. Find MAP for these queries.

2. Consider the following document by term binary D matrix for m= 6 documents (rows), n= 6 terms (columns).

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Consider the problem of constructing a document by document similarity, S, matrix. How many similarity coefficients will be calculated using the term inverted indexes (the third method, the most efficient one). How much do we save with this method with respect to the brute force, the first, method.

3. In this part consider the paper J. Zobel, A. Moffat, "Inverted files for text search engines." *ACM Computing Surveys*, Vol. 38, No. 2, 2006.

a. Understand the skipping concept as applied to the inverted index construction.

Assume that we have the following posting list for term a: <1, 2> <3, 5> <9, 4> <10, 3> <15, 3> <17, 4> <18, 3>, <22, 2> <24, 4> <33, 4> <38, 5> <43, 5> <55, 3><64, 2> <68, 4> <72, 5> <75, 1> <88, 2>.. The posting list indicates that term-a appears in d1 twice and in d3 five times, etc.

Assume that we have the following posting list for term-b: <12, 2> <25, 2> <45, 2> <66, 1>.

Consider the following conjunctive Boolean query: term-a and term-b. If no skipping is used how many comparisons do you have to find the intersection of these two lists?

Introduce a skip structure, draw the corresponding figure then give the number of comparisons involved to process the same query.

State the advantages and disadvantages of large and small skips in the posting lists. Note that in the paper it is assumed that compression will be used. The skip idea is applicable in an uncompressed environment too.

b. Give a posting list of of term-a (above it is given in standard sorted by document number order) in the following forms: 1), a) ordered by $f_{d,t}$, b) ordered by frequency information in prefix form. (Consider the J. Zobel, A. Moffat paper.)

i. Under which condition (if any) a is more preferable then b? OR Can we design a scenario that would make a better than b?

ii. Under which condition (if any) b is more preferable then a? OR Can we design a scenario that would make b better than a?

4. IR test collection:

a. What are the components of an information retrieval test collection?

b. Explain the pooling approach.

c. Please read the paper by Zobel (How Reliable Are the Results of Large-Scale Information Retrieval Experiments?) and give some reflections of his criticism of pooling.

5. For the matrix given in question 2 apply single-link and complete-link clustering algorithms. Use the Dice coefficient for obtaining the similarity matrix S.

6. Assume that we have a similarity matrix obtained by an asymmetric similarity measure: In such a matrix s_{ij} and s_{ji} may may have different values.

a. Find an asymmetric similarity measure from the web (different from the one, cover coefficient concept, we discuss in the class) and give the citation of the work that defines the measure, its definition, and a simple computation example.

b. Give a real life example that would require an asymmetric similarity measure.

c. Consider the dendrogram structure obtained by using the upper diagonal entries vs, the one obtained by lower diagonal entries. Is it meaningful to obtain two different clustering structures? Can these structures be different? If so what does it mean?

7. For the matrix given in question 2 apply C^3M and

a. Obtain the C matrix,

b. Explain how you select the cluster seeds,

c. Give/draw the IISD data structure,

d. Obtain the clusters: briefly explain how you assign non-seeds to seed documents.

8. For C^3M prove the following

a. if $c_{ij}= 0$ then $c_{ji}= 0$

b. $n_c = n_c'$ that is the number of clusters implied by documents is the same as terms.

9. Consider a data stream environment (tweets, news articles, etc.). In stream related applications data objects come one after the other in a temporal order, we can only keep the later objects in the memory due to storage limitation, we have to process the instances (objects) as they come, and importance of old objects drop as time passes.

a. Define four or more additional real life examples for such environments,
b. Define a dynamic clustering algorithm for an application of your choice (if you use an existing algorithm for inspiration give its citation),
c. Define its parameters and suggest ways of defining them,
d. Is there any hint for such an algorithm/environment in the paper by Jain, Murthy, and Flynn: Data Clustering: A Review, *ACM Computing Surveys*, 1999?